FIG. 1

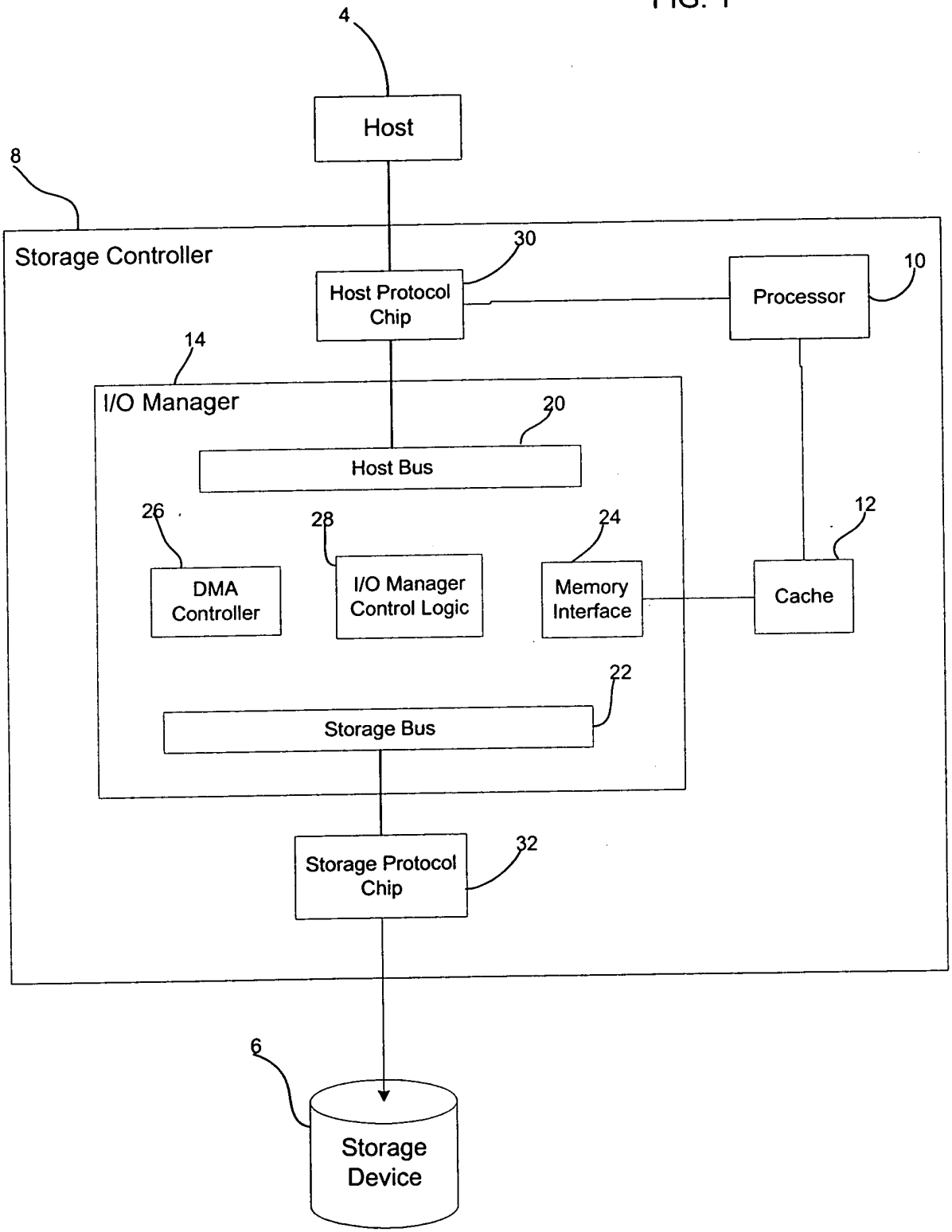

4
Host
8
Storage Controller
30
Host Protocol Chip
10
Processor
14
I/O Manager
20
Host Bus
26
DMA Controller
28
I/O Manager Control Logic
24
Memory Interface
12
Cache
22
Storage Bus
32
Storage Protocol Chip
6
Storage Device

FIG.2

50

Hardware Control Block	
Byte	**Field**
<u>0-5</u> 0 - 1 2 - 5	<u>Target PA</u> Disk ID LBA
6-7	Reserved for processor
8-23	Reserved for I/O Manager
<u>24-25</u> Bits 0 - 9	<u>Controls</u> Reserved
Bit 10	Recalculate LRC after Target PA is added and add LRC
Bit 11	Add target PA; use the target PA. The LBA is used for the first transferred sector. Increment the LBA for subsequent transferred sector.
Bit 12	Check LRC
Bit 13	Check PA. Use the source PA. LBA must match first transferred sector. Increment the LBA for subsequent transferred sector.
Bits 14 - 15	Address Conversion: 00 - 512 cache to 512 host 01 - 512 host to 520 cache 10 - 520 cache to 512 disk 11 - 520 cache/disk
<u>26-31</u> 26-27 28-31	<u>Source PA</u> Disk ID LBA

FIG.3

60

ADDRESS FORMAT	
Bit Offset	Field Description
63	Reserved
62	Hardware Control Block enabled (0 - do not use HCB; 1 - use HCB index
61-64	Hardware Control Block Index.
43-33	Reserved
32-0	Memory Address

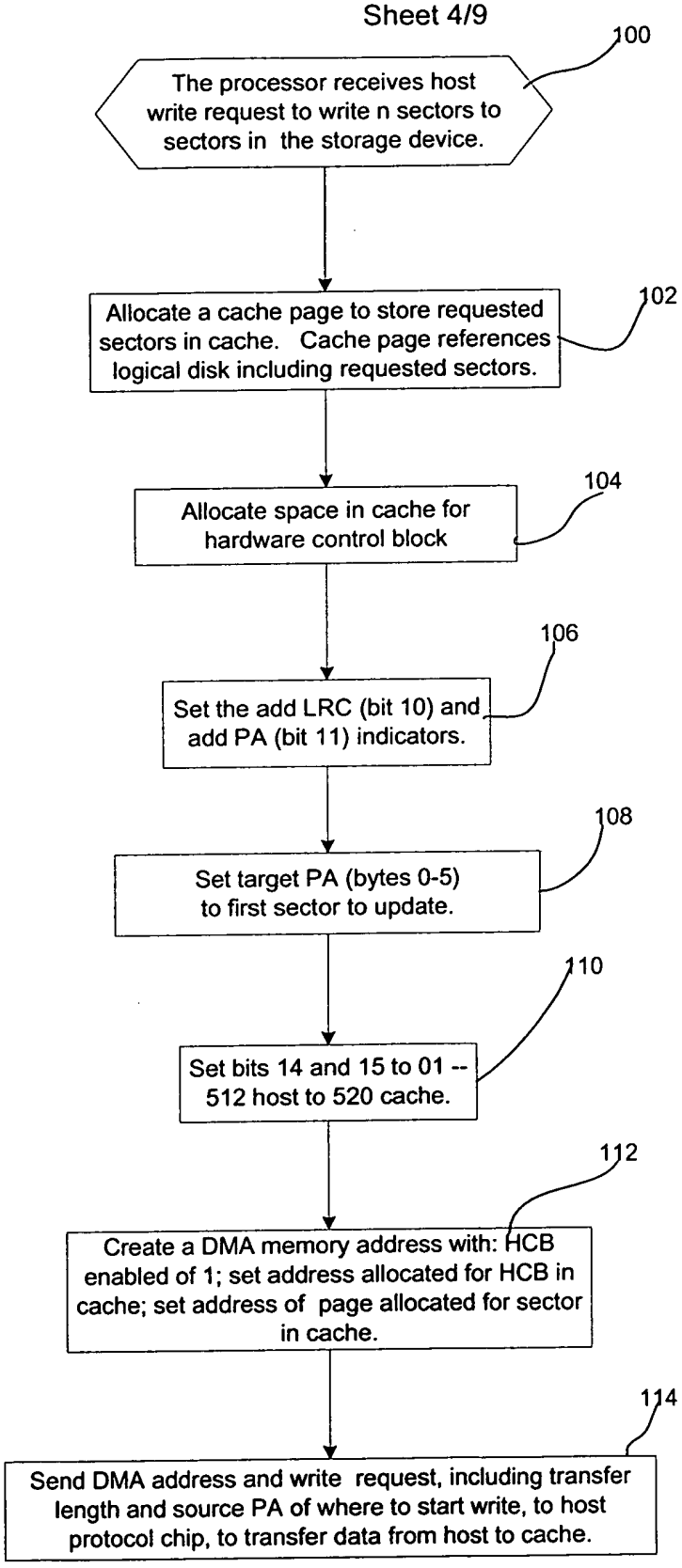


FIG.4

FIG.5

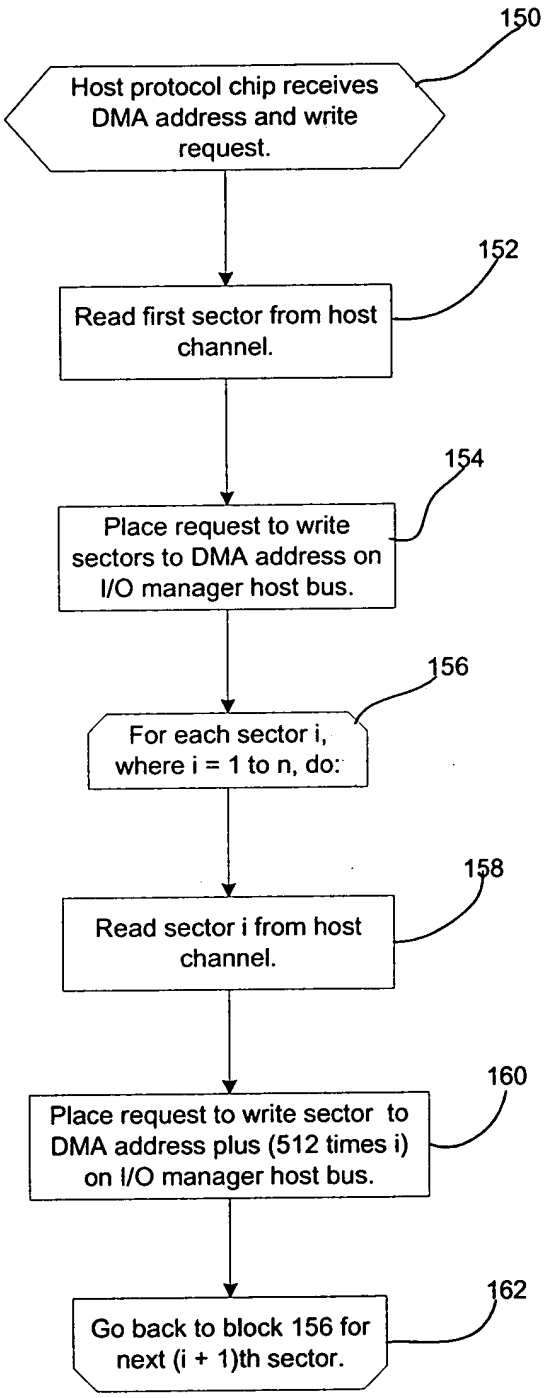

150
Host protocol chip receives DMA address and write request.
152
Read first sector from host channel.
154
Place request to write sectors to DMA address on I/O manager host bus.
156
For each sector i, where i = 1 to n, do:
158
Read sector i from host channel.
160
Place request to write sector to DMA address plus (512 times i) on I/O manager host bus.
162
Go back to block 156 for next (i + 1)th sector.

FIG.6

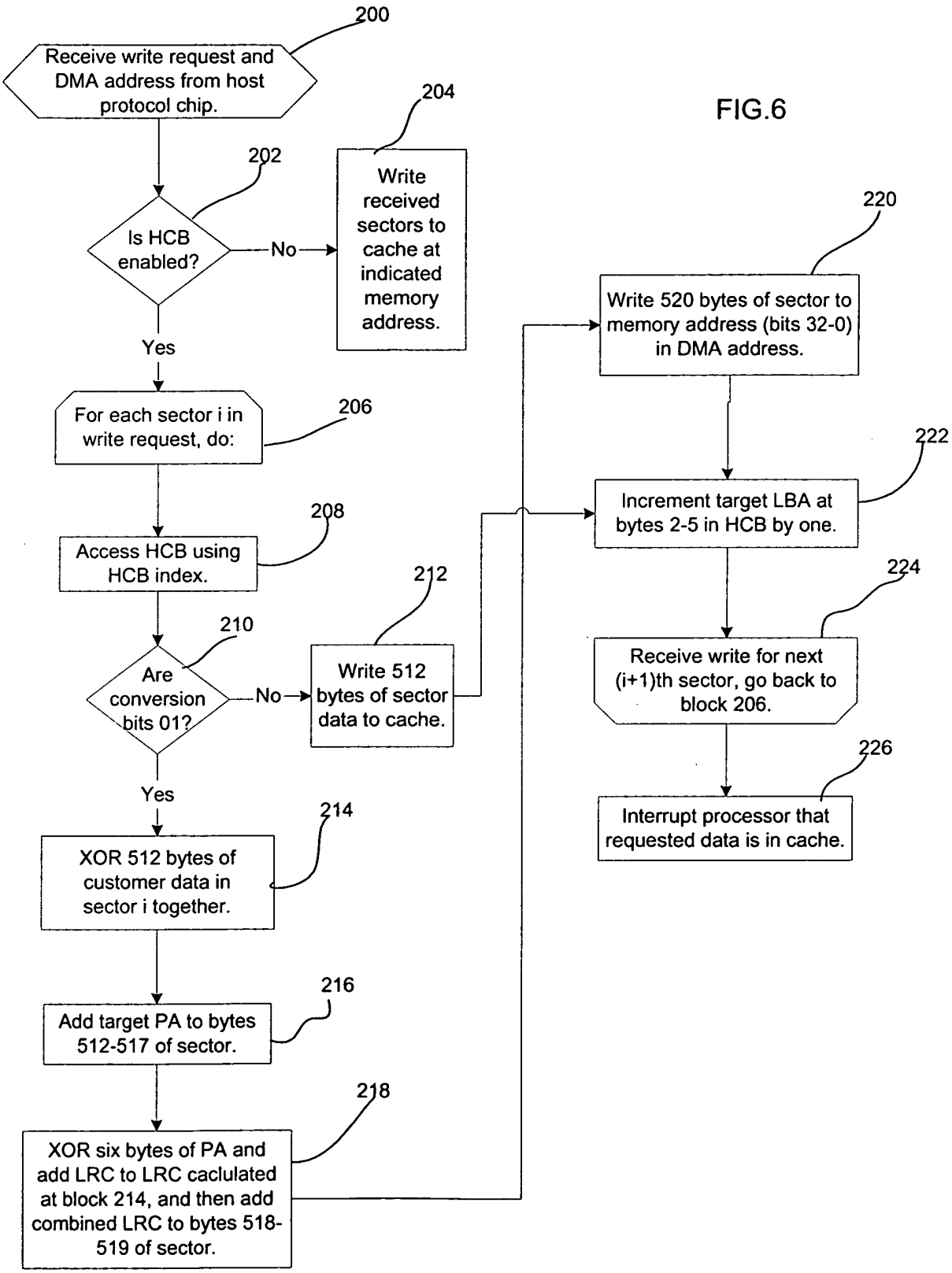

200
Receive write request and DMA address from host protocol chip.
202
Is HCB enabled?
No
204
Write received sectors to cache at indicated memory address.
Yes
206
For each sector i in write request, do:
208
Access HCB using HCB index.
210
Are conversion bits 01?
No
212
Write 512 bytes of sector data to cache.
Yes
214
XOR 512 bytes of customer data in sector i together.
216
Add target PA to bytes 512-517 of sector.
218
XOR six bytes of PA and add LRC to LRC caclulated at block 214, and then add combined LRC to bytes 518-519 of sector.
220
Write 520 bytes of sector to memory address (bits 32-0) in DMA address.
222
Increment target LBA at bytes 2-5 in HCB by one.
224
Receive write for next (i+1)th sector, go back to block 206.
226
Interrupt processor that requested data is in cache.

FIG.7

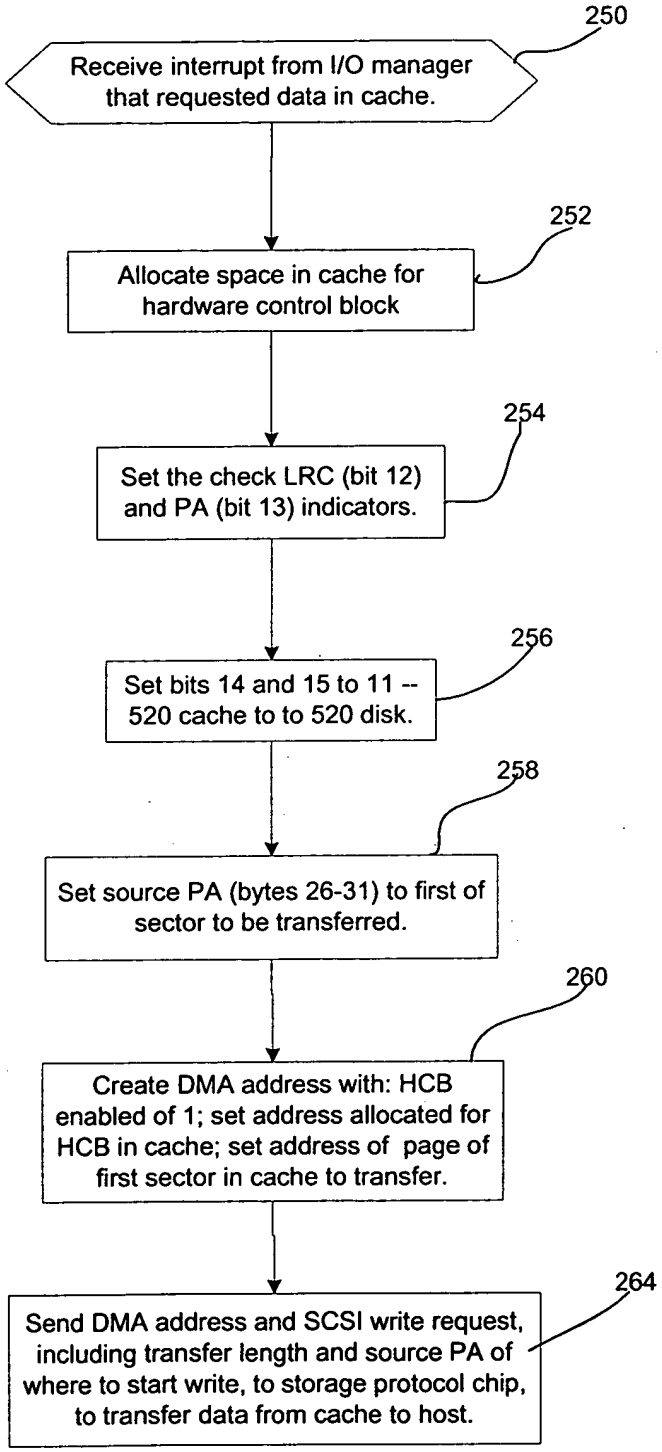
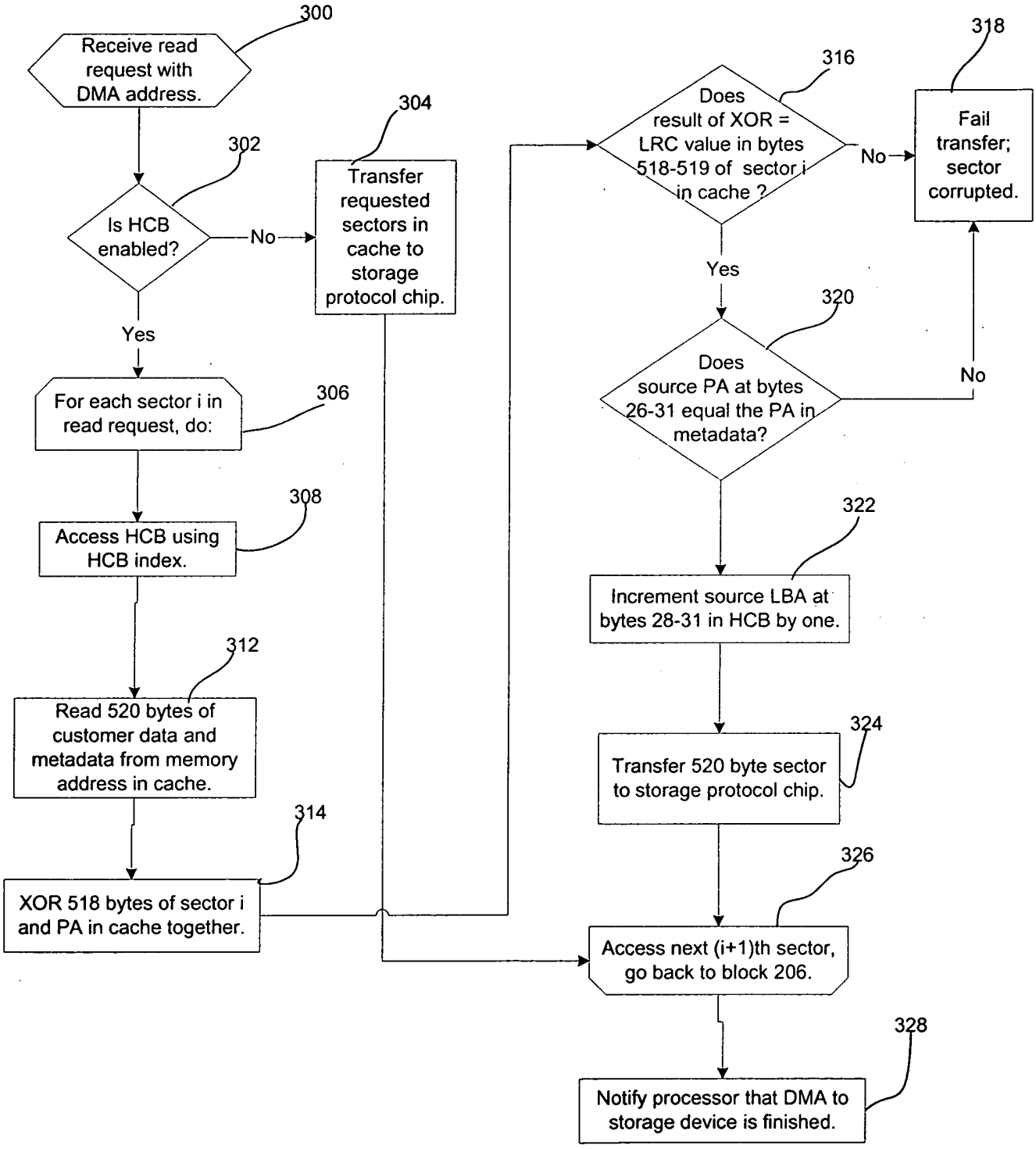

250
Receive interrupt from I/O manager that requested data in cache.
252
Allocate space in cache for hardware control block
254
Set the check LRC (bit 12) and PA (bit 13) indicators.
256
Set bits 14 and 15 to 11 -- 520 cache to to 520 disk.
258
Set source PA (bytes 26-31) to first of sector to be transferred.
260
Create DMA address with: HCB enabled of 1; set address allocated for HCB in cache; set address of page of first sector in cache to transfer.
264
Send DMA address and SCSI write request, including transfer length and source PA of where to start write, to storage protocol chip, to transfer data from cache to host.

FIG.8


300
Receive read request with DMA address.
302
Is HCB enabled?
No
304
Transfer requested sectors in cache to storage protocol chip.
Yes
For each sector i in read request, do:
306
Access HCB using HCB index.
308
Read 520 bytes of customer data and metadata from memory address in cache.
312
XOR 518 bytes of sector i and PA in cache together.
314
316
Does result of XOR = LRC value in bytes 518-519 of sector i in cache ?
No
318
Fail transfer; sector corrupted.
Yes
320
Does source PA at bytes 26-31 equal the PA in metadata?
No
322
Increment source LBA at bytes 28-31 in HCB by one.
324
Transfer 520 byte sector to storage protocol chip.
326
Access next (i+1)th sector, go back to block 206.
328
Notify processor that DMA to storage device is finished.

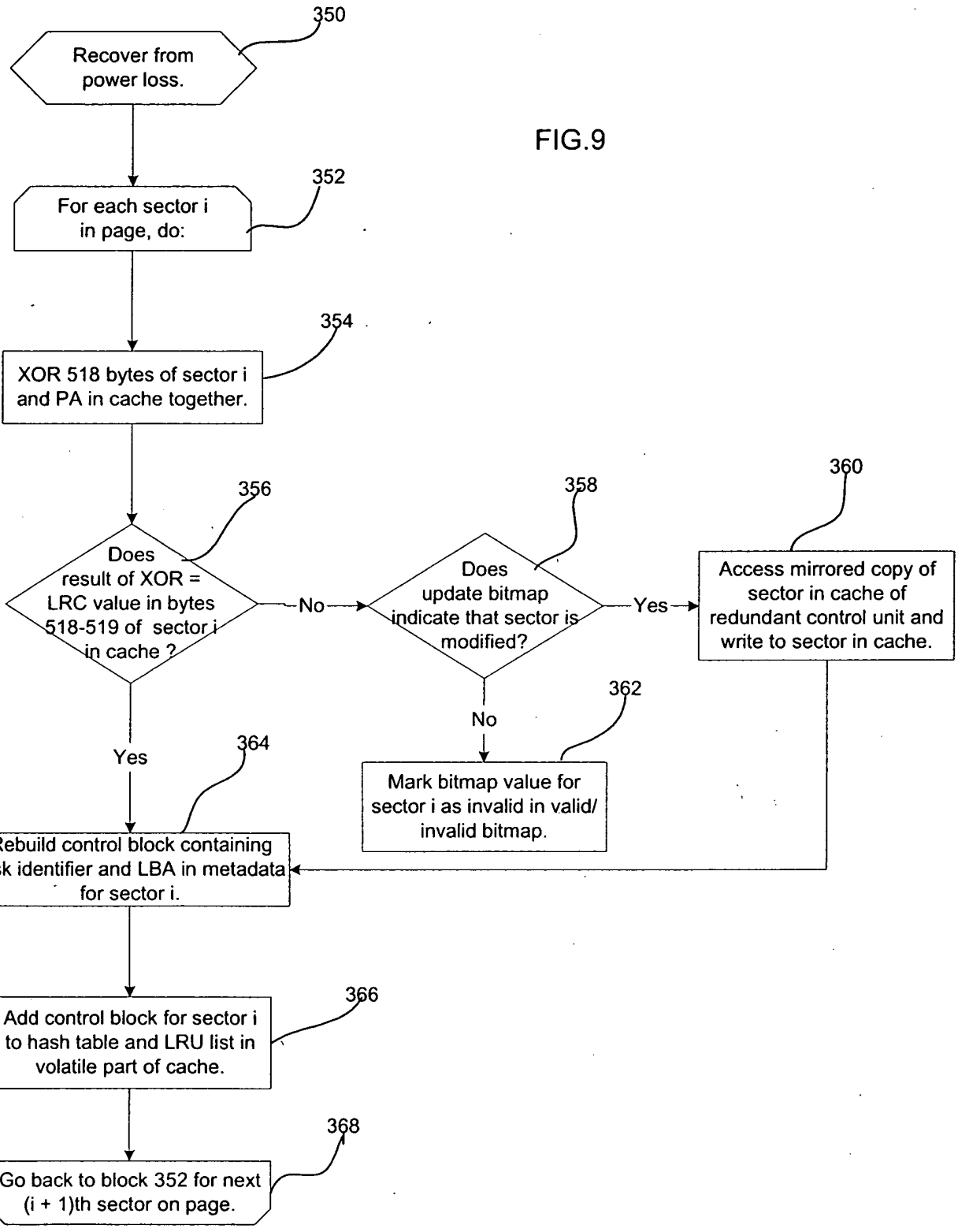

350
Recover from power loss.
352
For each sector i in page, do:
354
XOR 518 bytes of sector i and PA in cache together.
356
Does result of XOR = LRC value in bytes 518-519 of sector i in cache ?
No
358
Does update bitmap indicate that sector is modified?
Yes
360
Access mirrored copy of sector in cache of redundant control unit and write to sector in cache.
No
362
Mark bitmap value for sector i as invalid in valid/ invalid bitmap.
Yes
364
Rebuild control block containing disk identifier and LBA in metadata for sector i.
366
Add control block for sector i to hash table and LRU list in volatile part of cache.
368
Go back to block 352 for next (i + 1)th sector on page.

FIG.9